Master's thesis

Marginal pseudolikelihood in labelled graphical models

Taneli Pusa
013766221
University of Helsinki

April 28, 2015

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of science	Department of mathematics and statistics

Tekijä — Författare — Author
Taneli Pusa

Työn nimi — Arbetets titel — Title
Marginal pseudolikehood in labelled graphical models

Oppiaine — Läroämne — Subject
Mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	April 2015	32

Tiivistelmä — Referat — Abstract

The objective of this work is to generalize the pseudolikelihood-based inference method from ordinary Markov networks to an extension of the model containing context-specific independencies: the labelled graphical model. Probabilistic graphical models like the Markov and Bayes networks are used to represent the dependence structure of multivariate probability distributions. Machine learning methodology can then be used to learn these dependence structures from sample data. The Markov network is a model, which assigns no directionality to interactions between variables: the probability distribution is represented by an undirected graph, where nodes correspond to variables and edges to direct interactions. A labelled graphical model extends this idea by assigning labels to edges to represent contexts, i.e outcomes of other variables in the distribution, in which the associated variables are independent.

Bayesian inference can be used to learn the dependence structure of a set of variables using data. The standard procedure is to consider the posterior probability of a model given the data and aim to maximize this score. This involves explicitly calculating the marginal likelihood of the model. In the case of Markov networks and consequently labelled models, this can not be done analytically and approximation methods must be used. Pseudolikelihood is one such method, which allows for both the analytical calculation of the so-called marginal pseudolikehood replacing the actual marginal likelihood of a model and the computationally very advantageous property of a node-wise factorizable score-function.

This thesis presents the general theory behind the labelled graphical models and the basics of Bayesian inference. The pseudolikelihood approximation is introduced and applied to labelled models and the consistency of the score is proved. Lastly a greedy hill climb -algorithm is used to demonstrate the inference in practice by a synthetic and a real data example.

Avainsanat — Nyckelord — Keywords
Machine learning, Bayesian inference, Graphical models, Pseudolikelihood

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula campus library

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

Probabilistic graphical models are used to represent high dimensional distributions in a simple, compact way in a wide variety of applications spanning from physics to biology and sociology (see Lauritzen, 1996 and Koller and Friedman, 2009). Markov networks are a model class that uses an undirected graph to define the dependence structure of a multivariate distribution with no clear directionality in interactions between the variables. In some applications Markov networks' requirement of independence to be valid over the complete outcome space of the variables in question is too restricting. To address this, Corander (2003) introduced the idea of *labelled graphical models* (LGM), which allows context-specific independencies to be marked directly on the graph. However, since the statistical inference of the model structure is hard due to the intractability of the normalizing constant, both Corander (2003) and more recently Nyman et al. (2014) have restricted the model space to decomposable graphs. Here we aim at loosening this restriction by using the *marginal pseudolikelihood* (MPL) to determine the optimal model for data. In essence we try to combine the efforts of Pensar et al. (2014b) who proved the consistency of the MPL for ordinary Markov networks and showed that it is a valid candidate for a likelihood score and Pensar et al. (2014a) who investigated the context-specific generalization of the Bayes network, thus paving the way to extend the MPL to labelled models.

This thesis is structured as follows. In Section 2 we introduce most of the notation, the Markov network and its context-specific extension and investigate two parameterization schemes for these models. In Section 3 we briefly discuss the concept of Bayesian inference in general, introduce the pseudolikelihood approximation (MPL) for LGMs and prove its consistency and construct a method in the form on an algorithm to learn LGMs from data. In Section 4 synthetic and real data examples are provided and studied. In Section 5 some discussion and remarks are provided.

2 Graphical Models

Graphical models are used to represent complex dependence structures in a compact way. Their usefulness lies in their simplicity and intuitively clear interpretation. The distribution is represented by a graph, the nodes of which correspond to variables and edges to direct interactions between them. Thus, for example conditional independencies can be read directly of the graph. Also, graphs are a data type easily handled by a computer and efficient algorithms for manipulating them are readily available.

The theory of graphical models has its origins in log-linear and covariance selection models and can be traced back to Birch (1963, 1964), but it was the paper by Darroch

et al. (1980) that can be said to have started the modern development (Whittaker 1990). Since then a great deal of research effort has been put to investigating both theory and applications.

Two widely used classes of graphical models are the Bayes network and the Markov network. The former focuses on causal relationships between variables by attributing directionality to interactions. In this work we will focus on the latter, which assumes interactions to be directionless or at least doesn't explicitly state one variable to be the cause of another.

2.1 Markov Networks

A Markov network is an undirected graph representing the joint distribution of a set of variables. Its nodes represent the variables themselves and the edges dependencies amongst those variables. An edge between two nodes signifies a direct dependence between the associated variables and lack of one conditional independence. We will restrict our attention to discrete random variables with positive distributions.

Let $X = \{X_1, \ldots, X_d\}$ be a set of variables. The outcome space for variable X_i is then $\mathcal{X}_i$ and the joint outcome space of a set of variables $X_S = \{X_j : j \in S\}$ is the cartesian product $\mathcal{X}_S = \times_{j \in S} \mathcal{X}_j$. The cardinality of an outcome space is denoted by $|\mathcal{X}_S|$. A specific value taken by variable X_i is $x_i \in \mathcal{X}_i$ and for a set of variables $x_S \in \mathcal{X}_S$. These are indexed as $x_i^{(j)}$ and $x_S^{(k)}$, so that for example for a binary variable X_i, $j = 1, 2$. A data set $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is a sample obtained from the joint distribution of X, consisting of independent observations $\mathbf{x}_k = (x_{k_1}, \ldots, x_{k_d})$.

The conditional probability $p(X_i = x_i | X_j = x_j)$ will be shortened to $p(x_i|x_j)$ and consequently $p(X_i|x_j)$ will denote a conditional probability distribution and $p(X_i|X_j)$ a collection of these distributions. $X_i \perp X_j$ denotes the independence of two variables and $X_i \perp X_j | X_k$ expresses that X_i is independent of X_j given X_k that is $p(X_i|X_j, X_k) = p(X_i|X_k)$. All these generalize readily to sets of variables.

Definition 1. *A Markov network over a set of random variables X is defined as a graph $G = (V, E)$, where $V = \{1, \ldots, d\}$ is a set of nodes that corresponds to the indices of the variables and $E \subseteq \{V \times V\}$ is a set of edges. An edge between two nodes signifies a direct dependence between the corresponding variables. The set of all possible Markov networks over X is $\mathcal{G}$.*

We say that node i is a neighbour of j or adjacent to j – and vice versa – if $\{i, j\} \in E$. A set of neighbours $\{j \in V : \{i, j\} \in E\}$ is called a Markov blanket and node i's Markov blanket is denoted by $mb(i)$. The intersection of the Markov blankets of two nodes i and j – their common neighbours – is $P_{\{i,j\}}$. It should be noted that although for the sake of clarity, we will denote a variable with X_i and the corresponding node with i, the words

are in this context for most purposes interchangeable and will often be used as such.

A *clique* in a graph $G = (V, E)$ is a set of nodes $M \subseteq V$ such that $\{i, j\} \in E$ for all $i, j \in M$. That means that a clique is a subgraph of G, where every node is adjacent to each other. A *maximal clique* is a clique into which no node can be added. A *path* is an ordered set of nodes such that there is an edge between every consecutive node in the path. A set of nodes A is said to be *separated* from a set of nodes B by a set of nodes C if all paths from A to B contain at least one node in C. The Markov blanket of a node separates it from the rest of the graph. A *cycle* is a path from a node back to the same node. A *chord* is an edge between two non-consecutive nodes in a cycle. A graph G is considered *decomposable* if all cycles in G containing four or more unique nodes contain at least one chord. This concept is illustrated in Fig. 2.1 a) and b).

A Markov network reflects the dependence structure of X through a series of *Markov*

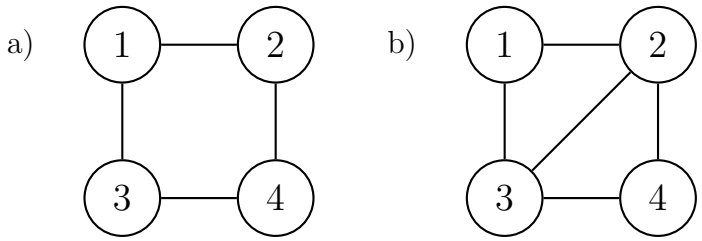

Figure 2.1: Graph in a) is not decomposable. Addition of edge $\{2, 3\}$ in b) ensures decomposability.

properties:

$$1)\ X_i \perp X_j | X_{V \setminus \{i,j\}} \qquad \forall \{i, j\} \notin E \tag{2.1}$$

$$2)\ X_i \perp X_{V \setminus \{mb(j) \cup i\}} | X_{mb(i)} \qquad \forall i \in V \tag{2.2}$$

$$3)\ X_A \perp X_B | X_C \qquad \text{for all disjoint subsets of V such that C separates A from B} \tag{2.3}$$

referred to as pairwise, local and global respectively. This feature illustrates the usefulness of Markov networks in encoding the qualitative nature of a multivariate distribution. Consider the graph in Fig. 2.2 a). We can immediately see for example that if we know the value of variable 2, knowing the values of 3 and 4 bring no new information about variable 1. This way even complicated dependence structures can be expressed in a compact way that is easily interpretable.

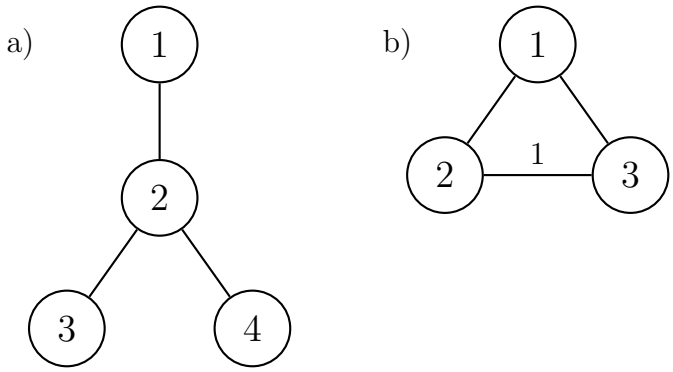

Figure 2.2: a) A simple graph representing the dependence structure of four variables. b) The label on edge $\{2,3\}$ signifies that $X_2 \perp X_3 | X_1 = 1$ and vice versa.

2.2 Context-specific Models

Although the original Markov network is a useful model largely because of its simplicity, it can be too "crude" to encode some intricate structures faithfully. Especially in the case where the variables have multiple possible values, dependence that is only present in a subset of outcomes can be overlooked. In order to overcome this problem we adopt the concept of labelled graphical models (LGM) presented by Corander (2003).

We say that X_i is independent of X_j in the context $X_k = x_k$ or

$$X_i \perp X_j | X_k = x_k$$

if $p(X_k|X_j, x_k) = p(X_k|x_k)$. In a situation, where this context-specific independence is present say for only one of the outcomes x_k, we would like to somehow include this information into our graphical model. This can be done by adding labels to the simple graph model presented in the previous section.

Definition 2. *A label on an edge $\{i,j\}$ is the set $L_{\{i,j\}} = \{x_{P_{\{i,j\}}} \in \mathcal{X}_{P_{\{i,j\}}} : X_i \perp X_j | X_{P_{\{i,j\}}} = x_{P_{\{i,j\}}}\}$. The set of all labels ascribed to a graph is called L and a labelled graph is $G_L = (V, E, L)$.*

Now a label on an edge expresses in which contexts the adjoined variables are actually independent. Note that if we hold on to a numeration of the variables (nodes), we need only write the joint outcome of the common neighbours and agree that the ordering follows the numerical ordering of the concerned variables (nodes). An example is given in Fig. 2.2 b).

A LGM is considered decomposable if no edge that is in the intersection of two maximal cliques has a label and if in no maximal clique there are labeled edges which share

no nodes. This is illustrated in Fig. 2.3. Decomposability of LGMs is important because it allows for the factorization of the underlying probability distribution according to the maximal cliques (Nyman et al. 2014).

An inherent problem with LGMs is model identifiability. It is generally not guaran-

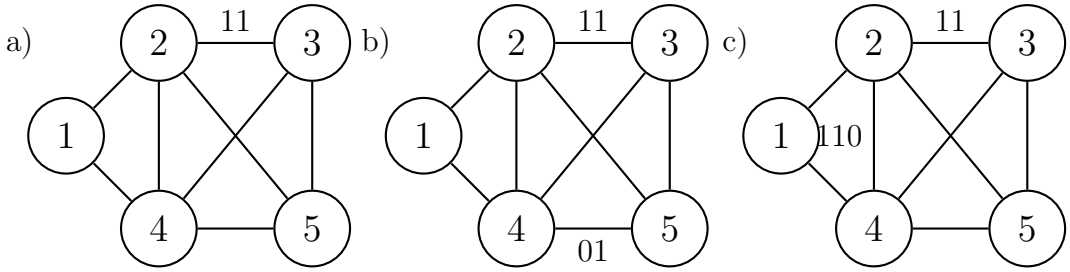

Figure 2.3: The LGM in a) is decomposable. The LGMs in b) and c) are not. In b) labelled edges in maximal clique $\{2, 3, 4, 5\}$ have no common nodes. In c) a label is placed on the edge $\{2, 4\}$ that is the intersection of maximal cliques $\{1, 2, 4\}$ and $\{2, 3, 4, 5\}$.

teed that the G_L for a distribution is unique (see Corander 2003 and Nyman et al. 2014). To account for this we introduce the *maximality* and *regularity* constraint.

Definition 3. *A labelled graph G_L is considered maximal and regular, if no label can be added to L without changing the associated distribution and if all labels $L_{\{i,j\}}$ are proper subsets of $\mathcal{X}_{P_{\{i,j\}}}$*

By requiring LGMs to be maximal and regular we can ensure that the correct model corresponding to the distribution in question is uniquely identified (see Corander 2003).

What the labels do is induce partitions to the outcome space $\mathcal{X}_{mb(i)}$, grouping together joint outcomes of neighbouring variables that are equivalent in terms of the conditional probabilities for the variable in question. The maximal and regular condition require that firstly no label can be added which would not induce a change to the partition and secondly that no label covers all outcomes so that in essence the edge to which it is added becomes vain. Since it is later needed, we will adopt a notation for keeping track of these partitions as well. We call $\mathcal{S}_i$ a partition of the outcome space $\mathcal{X}_{mb(i)}$ so that $S_i^{(j)}$ are sets of outcome indices and

$$p(X_i|x_{mb(i)}^{(k)}) = p(X_i|x_{mb(i)}^{(l)}) \quad \forall k, l \in S_i^{(j)}.$$

From now on we will refer to the set of possible maximal and regular LGMs over X as $\mathcal{G}_L$.

2.3 Parameterization

The simplest way to characterize a probability distribution is to state the probability for each different outcome. However, this probability vector is in no explicit way related to the graphical representation presented earlier. Here we look at two different parameterization schemes directly related to the graph structure representing the distribution.

2.3.1 Clique Factors

A factor is – in an intuitive sense – a measure of the *affinity* of two outcome values (Koller and Friedman 2009).

Definition 4. *Let X be a set of random variables. A factor is a function $\phi : \mathcal{X} \to \mathbb{R}$.*

We define probabilities by taking products of factors and normalizing:

$$p(x_1, x_2, x_3) = \frac{1}{Z}\phi(x_1, x_2)\phi(x_2, x_3)\phi(x_3, x_1), \quad Z = \sum_{x_{\{1,2,3\}} \in \mathcal{X}_{\{1,2,3\}}} \phi(x_1, x_2)\phi(x_2, x_3)\phi(x_3, x_1).$$

Z is often referred to as the *partition function*. Factors relate to the structural properties of a graph through cliques. A distribution whose independencies are consistent with those represented by a Markov network factorizes according to

$$p(X) = \frac{1}{Z}\phi(X_{S_1}) \dots \phi(X_{S_k}),$$

where $X_{S_i} (i = 1, \dots, k)$ are cliques in the graph. We can reduce the number of parameters by requiring every clique in X_{S_i} to be maximal (Koller and Friedman 2009). Maximal cliques are relevant to our interests since Nyman et al. (2014) showed that when decomposability is assumed, the marginal likelihood (see Section 3) of a LGM factorizes accordingly and is thus possible to calculate analytically.

2.3.2 Log-linear Expansion

A general reference for this section and the source of the following example is Whittaker (1990). A parameterization that most clearly illustrates the constraints imposed on a probability distribution by a LGM is the log-linear expansion. Consider first two binary random variables X_1 and X_2 taking values in $\{0, 1\}$. Their joint distribution can be expressed by

$$p(X_1, X_2) = p(0,0)^{(1-X_1)(1-X_2)} \cdot p(0,1)^{(1-X_1)X_2} \cdot p(1,0)^{X_1(1-X_2)} \cdot p(1,1)^{X_1X_2}.$$

Taking the logarithm and rearranging we get

$$\log p(X_1, X_2) = \log p(0,0) + X_1 \log\left(\frac{p(0,1)}{p(0,0)}\right) + X_2 \log\left(\frac{p(1,0)}{p(0,0)}\right) + X_1 X_2 \log\left(\frac{p(1,1)p(0,0)}{p(0,1)p(1,0)}\right).$$

We can now define the log-linear expansion of the joint distribution of these two variables as

$$\log p(X_1, X_2) = u_\varnothing + X_1 u_1 + X_2 u_2 + X_1 x_2 u_{12}$$

and more generally for a set of binary variables $X = \{X_1, \ldots, X_d\}$ as

$$\begin{aligned}\log p(X) &= u_\varnothing + \sum_i u_i X_i + \sum_{i,j} u_{ij} X_i X_j + \ldots + u_{12\ldots k} X_1 \ldots X_n \\ &= u_\varnothing + \sum_i u_i(X) + \sum_{i,j} u_{ij}(X) + \ldots + u_{12\ldots k}(X),\end{aligned}$$

where we take $u_a(X)$ to be zero whenever $X_i = 0$ for any $i \in a$. This can be further generalized to the multinomial case, but this we omit here. From now on we will leave out the argument X and remember that the only u-terms taken into account are those whose index is a subset of variables with non-zero values in the outcome in question with the exception of the u-term corresponding to the outcome $(0, \ldots, 0)$, which can be considered a normalizing factor of the parameterization. As an example, consider the two binary variables:

$$\begin{aligned}\log p(0,0) &= u_\varnothing \\ \log p(0,1) &= u_\varnothing + u_2 \\ \log p(0,1) &= u_\varnothing + u_1 \\ \log p(1,1) &= u_\varnothing + u_1 + u_2 + u_{12}.\end{aligned}$$

The relevance of the log-linear expansion in the context of graphical models is that the structure of the model is directly linked to the values of the u-terms. Consider first an ordinary Markov network. The absence of the edge $\{i, j\}$ expresses the statement $X_i \perp X_j | X_{V\setminus\{i,j\}}$, which means that the distribution factorizes according to

$$p(X) = p(X_i | X_{V\setminus\{i,j\}}) p(X_{V\setminus i}).$$

Taking the logarithm

$$\log p(X) = \log\left(p(X_i | X_{V\setminus\{i,j\}})\right) + \log\left(p(X_{V\setminus i})\right),$$

which means that the log-linear expansion must be of the form $f(X_{V\setminus j}) + g(X_{V\setminus i})$ and thus there can be no terms which depend on the values of both i and j. This means that all the u-terms which contain both variable indices in their index set must be zero.

Proposition 1. *If $X_i \perp X_j | X_{V\setminus\{i,j\}}$, in the log-linear expansion $u_a = 0$ for all a such that $\{i,j\} \in a$.*

A more rigorous proof is omitted and we refer the reader to Whittaker (1990) for a more detailed look.

Let us now move further and into the context-specific models. Once again the choice of log-linear expansion as a parameterization scheme is a very natural one. Consider the LGM in Fig. 2.2 b): from the graph we can read that $X_2 \perp X_3 | X_1 = 1$, from which it follows that

$$\begin{aligned} p(X_2 = 1 | X_1 = 1, X_3 = 0) &= p(X_2 = 1 | X_1 = 1, X_3 = 1) \iff \\ \frac{p(X_1 = 1, X_2 = 1, X_3 = 0)}{p(X_1 = 1, X_3 = 0)} &= \frac{p(X_1 = 1, X_2 = 1, X_3 = 1)}{p(X_1 = 1, X_3 = 1)} \iff \\ \frac{p(X_1 = 1, X_2 = 1, X_3 = 0) p(X_1 = 1, X_3 = 1)}{p(X_1 = 1, X_2 = 1, X_3 = 1) p(X_1 = 1, X_3 = 0)} &= 1. \end{aligned}$$

We get an identical equation for $X_2 = 0$ and combining these two:

$$\frac{p(X_1 = 1, X_2 = 1, X_3 = 0)}{p(X_1 = 1, X_2 = 1, X_3 = 1)} = \frac{p(X_1 = 1, X_2 = 0, X_3 = 0)}{p(X_1 = 1, X_2 = 0, X_3 = 1)},$$

which in terms of the log-linear expansion means

$$u_{23} + u_{123} = 0.$$

This is no coincidence, but actually an example of the way labels impose restrictions on the parameters.

Proposition 2. *Restrictions imposed on the log-linear expansion of a probability distribution by a label $L_{\{i,j\}}$ are of the form $\sum_a u_a = 0$, where a is a subset of variables containing both i and j and any number of variables with non-zero outcome in $L_{\{i,j\}}$.*

A proof for this proposition can be found in Corander (2003). We see that both removing edges and adding labels reduce the number of free parameters and in a sense simplify the distribution.

3 Bayesian Inference

Our ultimate goal for the remainder of this work is to establish means for using data to learn the dependence structure of a set random variables. We will do this using Bayesian inference: the method of evaluating each possible candidate hypothesis based on how well it predicts our observations. In this section we use this approach to device a scoring function for evaluating LGMs: the marginal pseudolikelihood. We start by a short introduction to Bayesian methodology via an example.

3.1 Prior and posterior distributions

The rationale behind the Bayesian approach to inference is illustrated by the following example: imagine you perform a coin tossing experiment by flipping a coin ten times. The resulting data set is

$$\mathbf{X} = (H, T, H, H, H, T, H, H, T, H)$$

"H" and "T" denoting "heads" and "tails" respectively. Each toss is a Bernoulli trial with

$$\begin{cases} p(\text{"heads"}) = q \\ p(\text{"tails"}) = 1 - q \end{cases}, \; q \in [0, 1].$$

Alternatively, the whole experiment is distributed according to the binomial distribution with the number of heads being the successes. For our purposes however, it is more convenient to consider each toss individually. Based on this data you now have to *infer* what is the value of $q \in [0, 1]$ and decide if the coin is biased or not.

One way to do this is to look at the *likelihood* of any specific parameter value q in light of the data. The likelihood of $q \in [0, 1]$ is defined as the probability of observing said data given that parameter value:

$$l(q|\mathbf{X}) = p(\mathbf{X}|q) = q^7(1 - q)^3.$$

This expression is maximized when $q = 7/10$, that is the frequency of heads in the data. This method of taking the most likely value of the parameter given the data is called the *maximum likelihood estimate* (MLE). However, you would not actually rule the coin as biased just based on this information alone, even though the value of q given by the estimate would suggest so. Now imagine your data had looked like this

$$(H, H, H, H, H, H, H, H, H, H).$$

Would you now state that the probability of the coin landing tails up is zero, that is $q = 1$? After all, tossing heads ten times in a row with a normal coin is unlikely, but not at all unheard of. Furthermore, imagine you know, based on the physical properties of the coin that there must be at least some small chance of it landing tails up.

In Bayesian inference these *a priori* beliefs about the parameters are quantified using *prior distributions*. For example in the coin tossing experiment we would like to somehow include in our analysis our prior knowledge that usually coins are almost exactly fair and that the probability of the coin being extremely biased is very low. We can do this by using the beta distribution, which is defined as

$$f(q, \alpha_1, \alpha_2) = \begin{cases} \frac{1}{B(\alpha_1, \alpha_2)} q^{\alpha_1}(1 - q)^{\alpha_2 - 1}, & q \in]0, 1[\\ 0, \quad \text{elsewhere}, \end{cases}$$

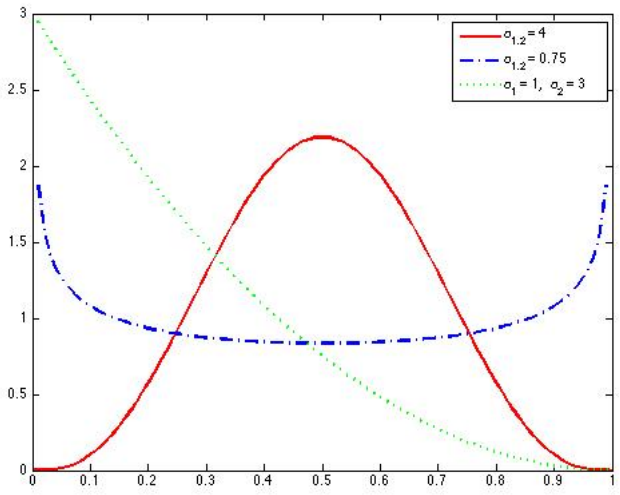

Figure 3.1: Beta distributions with different parameter values. The red solid line corresponds to our coin tossing example with most of the probability mass situated near the center.

where $B(\alpha_1, \alpha_2)$ is the beta function defined using the gamma function as

$$B(\alpha_1, \alpha_2) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1 + \alpha_2)}.$$

The α_1 and α_2 are *hyperparameters* determining the shape of the distribution. Fig. 3.1 shows a couple of examples. The red solid line in Fig. 3.1 shows the distribution when $\alpha_1 = \alpha_2 = 4$, which could be an example of the kind of prior we would use in the coin tossing experiment. It reflects our belief that the coin should be fair or only moderately biased by placing most the probability near the center. The hyperparameter values are equal making the distribution symmetric, meaning that we have no prior preference for one face over the other. The next example with $\alpha_1 = \alpha_2 = 0.75$ plotted with blue dash-dotted line shows another symmetric distribution, but this time one in favour of a bias towards one of the faces. The last example shows a distribution with $\alpha_1 = 1$ and $\alpha_2 = 3$. Now the distribution is not symmetric anymore, meaning we actually assume that if the coin is biased, it will be biased towards heads.

Let us now take the $f(q, 4, 4)$ as our prior distribution for the parameter q in the coin tossing example. We can now use the Bayes' rule (see for example Gelman et al. 2013)

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

to determine the *posterior distribution* of the parameter given our original set of data:

$$f(q|\mathbf{X}) = \frac{p(\mathbf{X}|q)f(q)}{p(\mathbf{X})} = \frac{\frac{1}{B(4,4)}q^7(1-q)^3q^{4-1}(1-q)^{4-1}}{\int_0^1 \frac{1}{B(4,4)}q^7(1-q)^3q^{4-1}(1-q)^{4-1}dq} = \frac{q^{10}(1-q)^6}{B(11,7)},$$

where the last step follows from the definition of the beta function. From the above expression we notice an interesting fact: it is of the same form as our prior distribution. Because of this property, we call the beta distribution a *conjugate prior* for the Bernoulli and binomial distributions. It means that when taken as a prior for the parameter values in said distributions, the posterior will always be of the same form as the prior. This is a very useful property since it makes calculating the posterior much easier.

This example demonstrates the Bayesian idea of updating the prior beliefs based on gathered information. Another interpretation which can give further intuition into the role of the hyperparameters is to look a the *maximum a posteriori estimate* (MAP), that is to maximize the posterior distribution. This is achieved with $q = 10/16$. We can now think of the hyperparameters as pseudo-observations or pseudocounts as they are often called: taking a prior which favoured values of q that correspond to a moderately biased coin is equivalent to adding synthetic observations with the same effect into our data set. This obviously only works for integer values of hyperparamaters, but the logic remains the same. Thus a prior can also be seen as a stabilizer, balancing the inference against inevitable noise in the data.

More generally for a multinomial random variable X taking values $(x_1, \ldots, x_k)$ with probabilities $(\theta_1, \ldots, \theta_k)$ we can define a prior with

$$f(\theta_1, \ldots, \theta_k, \alpha_1, \ldots, \alpha_k) = \begin{cases} \frac{1}{B(\alpha_1,\ldots,\alpha_k)} \prod_{i=1}^k x_i^{\alpha_i - 1}, & \theta_1, \ldots, \theta_k > 0, \sum_{i=1}^k \theta_i = 1 \\ 0, \quad \text{elsewhere.} \end{cases},$$

where B now denotes the multivariate beta function

$$B(\alpha_1, \ldots, \alpha_k) = \frac{\prod_{i=1}^k \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^k \alpha_i)}.$$

This is called the Dirichlet distribution and it is the conjugate prior for the multinomial distribution. It is a straightforward extension of the beta distribution to multiple dimensions. Note that there are only $k-1$ free variables since it has to hold that $\sum \theta_i = 1$ and thus $\theta_k = 1 - \sum^{k-1} \theta_i$. An equivalent definition with only $k-1$ variables is indeed often used.

3.2 Marginal likelihood

We now return to graphical models. Our goal is to find a suitable function to score each candidate model for a given set of variables. In contrast with the coin tossing example

given in the previous section, we do not wish to infer the exact values of the parameters. Instead we are interested in the qualitative nature of the underlying distribution as represented by a graph.

This turns the problem from a parameter estimation one to that of model comparison. We can think of the possible graphs for a set of variables as alternative hypotheses, each representing a particular dependence structure and our goal is to find which explains the observations best. For this task we need a method for evaluating the "fit" of a model to the data.

The Bayesian choice of course is the posterior probability of the graph given the data:

$$p(G|\mathbf{X}) = \frac{p(\mathbf{X}|G)p(G)}{p(\mathbf{X})}. \tag{3.1}$$

Because naturally the data and hence the denominator will always be the same for every model, we can focus on the numerator. The $p(\mathbf{X}|G)$ in the above expression is the *marginal likelihood* of the graph given the data defined as

$$p(\mathbf{X}|G) = \int_{\theta \in \Theta_G} l(\theta|\mathbf{X}, G) \cdot f(\theta|G) d\theta, \tag{3.2}$$

where $l(\theta|\mathbf{X}, G)$ is the likelihood of a specific parameter vector θ. To evaluate the marginal likelihood of G, we integrate over the set of all possible parameter values Θ_G (remember from Section 2.3 that the structure of G imposes restrictions on the parameters) weighed by some prior distribution over the parameter space, which will depend on the structure of the graph.

The marginal likelihood has the useful property that it acts as a natural Occam's razor: our wish is, for both aesthetical and practical reasons, to have simple models. For example in the case of graphs, one could say that every possible observation is explained by a complicated graph where every node is connected to one another. However, this would not be a very useful model. From Section 2.3 we know that the more complicated a dependence structure is, the more free parameters it has. Thus, in (3.2), a complex model has to spread its probability mass more widely and in doing so it loses its predictive power. For this reason (3.2) will favour simpler models (MacKay 2003). After all, we are trying to extract knowledge from the data and get a better understanding of the behaviour of our variables. An overtly complicated model will not succeed in these tasks.

Establishing prior distributions is central to Bayesian inference: without them the posterior cannot be evaluated. However, we can be as indecisive in their choice as we wish. If we do not want to express any *a priori* preferences about the parameter values, we can choose what is called a non-informative or objective prior. Intuitively this means that the distribution $f(\theta|G)$ is one which makes no statement about where the value of θ should

lie (for more information, see for example Kass and Wasserman, 1996). Furthermore, we can choose to completely ignore the prior probability of the model $p(G)$ by taking it to be uniform over all possibilities. Alternatively, we can choose to promote simpler models also through $p(G)$, in addition to the "natural" advantage they have in terms of the marginal likelihood.

With (3.1) as a function to score each candidate model, the task of finding the correct graph can be solved by (in principle) simple optimization. Unfortunately however, the marginal likelihood of a general Markov network cannot be calculated analytically (Pensar et al. 2014b). For this reason we must seek ways to approximate (3.2) somehow.

3.3 Pseudolikelihood

To overcome the problem of unavailable marginal likelihood we turn to the *pseudolikelihood* introduced by Besag (1972). This is a way to approximate the likelihood of a given parameter vector in the following way

$$pl(\theta|\mathbf{X}, G) = \prod_{i=1}^{d} p(\mathbf{X}_i|\mathbf{X}_{V\setminus i}, \theta). \tag{3.3}$$

Now remembering the Markov property (2.2), we can write this as

$$pl(\theta|\mathbf{X}, G) = \prod_{i=1}^{d} p(\mathbf{X}_i|\mathbf{X}_{mb(i)}, \theta). \tag{3.4}$$

We see that this is actually equivalent to considering the local structure around each variable separately as a Bayes network consisting only of the variable and its Markov blanket, with all the edges directed towards the corresponding node. The idea is illustrated in Fig. 3.2. This can be thought of as an interpretation of the Markov network not as undirected, but actually bi-directed, each edge representing a two-way influence between variables. Of interest here is that Pensar et al. (2014b) showed that using (3.3) in place of the likelihood function, the so-called *marginal pseudolikelihood* (MPL)

$$mpl(G|\mathbf{X}) = \hat{p}(\mathbf{X}|G) = \int_{\theta\in\Theta_G} pl(\theta|\mathbf{X}, G) \cdot f(\theta|G)d\theta \tag{3.5}$$

can be calculated analytically. Furthermore, they showed this to be enjoying consistency in the sense that it will eventually identify the correct graph structure as the sample size – that is the size of the data set in terms of observations – n tends to infinity.

From a more practical point of view, the MPL offers other perks as well. Besides consistency, Pensar et al. (2014b) demonstrated that it is both computationally efficient

and fares well against other similar methods such as using conditional mutual approach (Tsamardinos et al. 2003) and L_1-regularized logistic regression (Ravikumar et al. 2010). Moreover, unlike scores which explicitly punish complex graphs based on some preassigned function, the MPL does this through a prior and is therefore not as dependent on the correct choice of some tuning parameter.

Using our knowledge about the MPL as a scoring function based on the local dependence structure of each variable as a labelled Bayes network, we can easily extend it to LGMs following the treatment of Pensar et al. (2014a) who considered similar types of models and extended the actual marginal likelihood of a Bayes network to labelled acyclic graphs. Define the variables

$$n_{ijl} = \left| \{ \mathbf{x}_k \in \mathbf{X} : x_{k_j} = x_j^{(i)}, x_{k_{mb(j)}} \in S_j^{(l)} \} \right|,$$

which count the occurrence of specific joint outcomes of a variable and its Markov blanket in the data set $\mathbf{X}$ and

$$\theta_{ijl} = p(X_j = x_j^{(i)} | X_{mb(j)} = x_{mb(j)}^{(l)}),$$

which are the corresponding conditional probabilities. Let $r_j = |\mathcal{X}_j|$, $q_j = |\mathcal{X}_{mb(j)}|$ and $s_j = |\mathcal{S}_j|$. We now have for a LGM

$$pl(\theta | \mathbf{X}, G_L) = \prod_{j=1}^{d} \prod_{l=1}^{s_j} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}.$$

The notation becomes very heavy here, but the above basically expresses the likelihood of a specific parameter vector as a product of the probabilities defined by that vector, taking into account how many times an outcome whose probability we are considering appears in the data. If we take our prior distribution over the parameter space to be consisting of

Dirichlet distributions, it follows that

$$\begin{aligned}
\hat{p}(\mathbf{X}|G) &= \int_{\theta \in \Theta_{G_L}} pl(\theta|\mathbf{X}, G) \cdot f(\theta|G_L) d\theta \\
&= \prod_{j=1}^{d} \prod_{l=1}^{s_j} \int_{\theta_{jl}} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}} \frac{\prod_{i=1}^{r_j} \theta_{ijl}^{\alpha_{ijl}-1}}{B(\alpha_{1jl}, \ldots, \alpha_{r_j jl})} d\theta_{jl} \\
&= \prod_{j=1}^{d} \prod_{l=1}^{s_j} \int_{\theta_{jl}} \frac{\prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}+\alpha_{ijl}-1}}{B(\alpha_{1jl}, \ldots, \alpha_{r_j jl})} d\theta_{jl} \\
&= \prod_{j=1}^{d} \prod_{l=1}^{s_j} \frac{\Gamma(\alpha_{jl})}{\prod_{i=1}^{r_j} \Gamma(\alpha_{ijl})} \frac{\prod_{i=1}^{r_j} \Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(n_{jl} + \alpha_{jl})} \\
&= \prod_{j=1}^{d} \prod_{l=1}^{s_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl} + \alpha_{jl})} \prod_{i=1}^{r_j} \frac{\Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}
\end{aligned} \tag{3.6}$$

where α_{ijl} are hyperparameters associated with our Dirichlet distributions, $n_{jl} = \sum_{i=1}^{r_j} n_{ijl}$ and $\alpha_{jl} = \sum_{i=1}^{r_j} \alpha_{ijl}$. The integrals are taken over simplexes $\{\theta_{jl} = (\theta_{1jl}, \ldots, \theta_{r_j jl}) : \theta_{ijl} \geq 0, \sum_i \theta_{ijl} = 1\}$, from which the third step follows.

We choose our prior in accordance with Pensar et al. (2014a) who in turn used a modified version the non-informative prior of Buntine (1991):

$$\alpha_{ijl} = \frac{N \cdot |S_j^{(l)}|}{r_j \cdot q_j}$$

where N is the *equivalent sample size* expressing our prior belief in the uniformity of the conditional distributions coded by the graph. Although its value can affect the outcome of the inference (Silander et al. 2007), it is not paid too much attention here and from now on assumed to be 1.

The numerator in (3.1) contains another term besides the marginal likelihood – now replaced by the MPL: the prior probability of the graph structure. Often a uniform prior is selected and the term is thus omitted, but in the case of labelled models it can play an important role and in fact Pensar et al. (2014a) argue that without it, overfitting may occur. The downside is that the prior brings about a parameter which has to be determined, $\kappa \in]0, 1]$ which regulates how strongly an independence statement issued by a label has to be supported by the data in order for the label to be added. The prior is defined as

$$p(G_L) \propto \kappa^{dim(\Theta_G) - dim(\Theta_{G_L})} = \prod_{j=1}^{d} \kappa^{(q_j - s_j)(r_j - 1)},$$

$dim(\Theta_G)$ and $dim(\Theta_{G_L})$ being the number of free parameters in the underlying graph and the LGM respectively. Note that as the sample size increases, its effect will diminish. Also, the smaller the κ, the more "costlier" addition of labels will be while $\kappa = 1$ corresponds to a uniform prior.

In practice, instead of the marginal likelihood – or in this case pseudolikelihood – the logarithm of this quantity is used for several reasons. Firstly, it turns the product in (3.6) into a sum making it easier to handle and avoids problems that arise from too small or large numbers. Secondly, the logarithm of gamma function is widely available in computational software. From the theoretical point of view this brings no change since the model maximizing the logarithm will be the one maximizing the original score. We will denote the logarithm of MPL with logMPL.

We end this section with what is perhaps our most important statement: that of the consistency of the MPL score for LGMs.

Theorem 1. *Let $G_L^* = \{V^*, E^*, L^*\}$ be regular and maximal and define the true labelled graph structure of a LGM over variables $\{X_1, \cdots, X_d\}$. Let $\theta_{G_L^*} \in \Theta_{G_L^*}$ be a parameter vector defining a joint distribution to which G_L^* is faithful and from which a sample $\boldsymbol{X}$ of size n is obtained. The MPL estimator*

$$\widehat{G_L} = \underset{G_L \in \mathcal{G}_L}{\arg\max} \log\left(\hat{p}(\boldsymbol{X}|G)p(G_L)\right)$$

is consistent in the sense that $\widehat{G_L} = G_L^*$ eventually almost surely as $n \to \infty$.

Proof. First we remark that as its effect vanishes as $n \to \infty$, we can omit the prior probability of the graph $p(G_L)$. We note that

$$\log \hat{p}(\mathbf{X}|G) = \sum_{j=1}^{d}\sum_{l=1}^{k_j}\left[\log\Gamma\left(\alpha_{jl}\right) - \log\Gamma\left(n_{jl} + \alpha_{jl}\right) + \sum_{i=1}^{r_j}\left(\log\Gamma\left(n_{ijl} + \alpha_{ijl}\right) - \log\Gamma\left(\alpha_{ijl}\right)\right)\right]$$

for a graph without any labels is equivalent to the logMPL of an ordinary Markov network as considered by Pensar et al. (2014b) since s_j is now equal to q_j and $|S_j^{(l)}| = 1$ for all j and l. They showed that MPL is consistent in the sense that it will almost surely identify the Markov blankets for each variable as $n \to \infty$. Because the Markov blankets uniquely define a graph structure, the correct underlying graph will eventually be discovered. Furthermore, because the logMPL for a single variable is equivalent to the actual log-likelihood of a labelled Bayes network, also the correct label structure will be found. Since we assumed all models to be maximal and regular, as $n \to \infty$, $\widehat{G_L} = G_L^*$ almost surely. □

Though not crucial for the proof, a look at the asymptotic behaviour of the score is provided in appendix A.

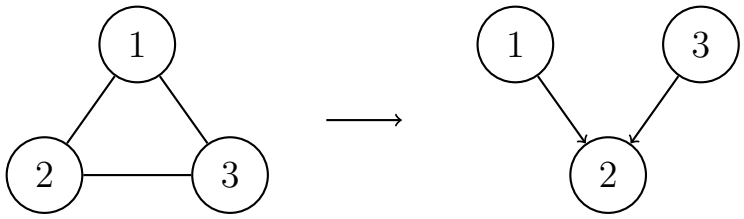

Figure 3.2: The original Markov network and a Bayes network consisting of variable 2 and its Markov blanket. These local structures for each variable are then combined in the final score.

3.4 Structure Learning

With the MPL as our score-function, we can now move on to discuss the problem of actually deciphering a graph structure from data in practice. That is, we wish to have an algorithm for finding the

$$\widehat{G_L} = \underset{G_L \in \mathcal{G}_L}{\arg\max} \log\left(\hat{p}(\mathbf{X}|G)p(G_L)\right)$$

Even with ordinary Markov networks, the size of the model space grows exponentially with the number of variables and adding labels further worsens the situation making an exhaustive search impossible in most relevant applications. The computational task of searching the model space is thus by far not a trivial one. However, since our primary interest here is the concept of using MPL for LGMs in the first place, we will restrict ourselves to relatively small models and use a simple greedy hill-climb approach. Ways to improve the efficiency of the algorithm are briefly discussed at the end of this section.

Since one of the advantages of including labels in our model is to be able to consider dependencies which are only present in a subset of outcomes, we would like to include the concept of labels already in the search for the underlying graph structure as opposed to for example discovering a Markov network and then optimizing the label structure of that set network.

We start with an empty set of edges E, an empty set of labels V and a set of possible edges E_{pos}. On each iteration of the algorithm, every possible single edge -change of the underlying graph is considered as follows: if the edge is present in E_{pos} and not in E, a candidate structure is constructed by adding the edge to the current graph and then optimizing the label structure of this new graph. For every edge that is already present in E, a candidate where said edge is removed is made and the label optimization again performed. Each candidate is then evaluated using the logMPL-score and the one improving the score the most compared to the current structure is chosen as the current LGM. This is continued as long as any improvement is acquired. The label optimization works much

the same way: for every edge in the graph, all possible changes to the label are assessed one by one, and the best in terms of logMPL-score is chosen as the new label. After every change the label is made maximal and regular to ensure consistency. This is continued as long as any improvement is made. In reality, the label optimization after each change in the underlying graph need not be done for the whole graph, since the structure only changes locally and thus the optimal label structure for nodes not concerned doesn't change. This feature once more highlights the benefits of a node-wise factorizable score. A pseudocode for the algorithm is provided in appendix B.

One immediately recognizes the aforementioned as a very computationally costly procedure and for bigger problems with more variables and outcomes, improvements are required. One possibility is to somehow prune the model space. In our approach, we used the complete set of edges $\{E \times E\}$ as the set of possible edges. This could be reduced by first approximating possible connections loosely to acquire a crude picture of the model. In particular, Pensar et al. (2014b) used the logMPL to search for Markov blankets of individual nodes in parallel, thus acquiring a set of possibly inconsistent Markov blankets that could then be combined. The prior pruning in our case could then consist of similar search, the set of possible edges comprising of edges discovered in at least one direction in the mb-search, and possibly with a looser criteria for adding edges in order to account for possible context-specific independencies. Stochastic algorithms, for example a Markov-chain Monte Carlo (MCMC) method similar to Pensar et al. (2014a) could also be used to further reduce the running time.

4 Examples

In this section we investigate some examples to illustrate the properties of our algorithm and the MPL-score. First we observe how the algorithm performs on a synthetic data set generated from a distribution faithful to a LGM shown in Fig. 4.1 a). To obtain the data, a probability vector describing the joint distribution was randomly generated by first drawing clique factors for the underlying graph from a normal distribution. The distribution thus obtained was then turned into a log-linear parameterization and the parameters were adjusted according to the restrictions imposed by the labels in a recursive fashion. This rather cumbersome procedure was chosen because it turned out that simply drawing the log-linear parameters from a normal distribution often resulted in pathological distributions with very weak dependencies. Finally the log-linear parameterization was turned back into a probability vector and sampled directly.

Fig. 4.1 b) - d) shows some examples of how different parameter values affect the outcome of the search procedure. With κ – the parameter defining our prior over the model space – equal to 0.3, the value determined best by Pensar et al. (2014a), we usu-

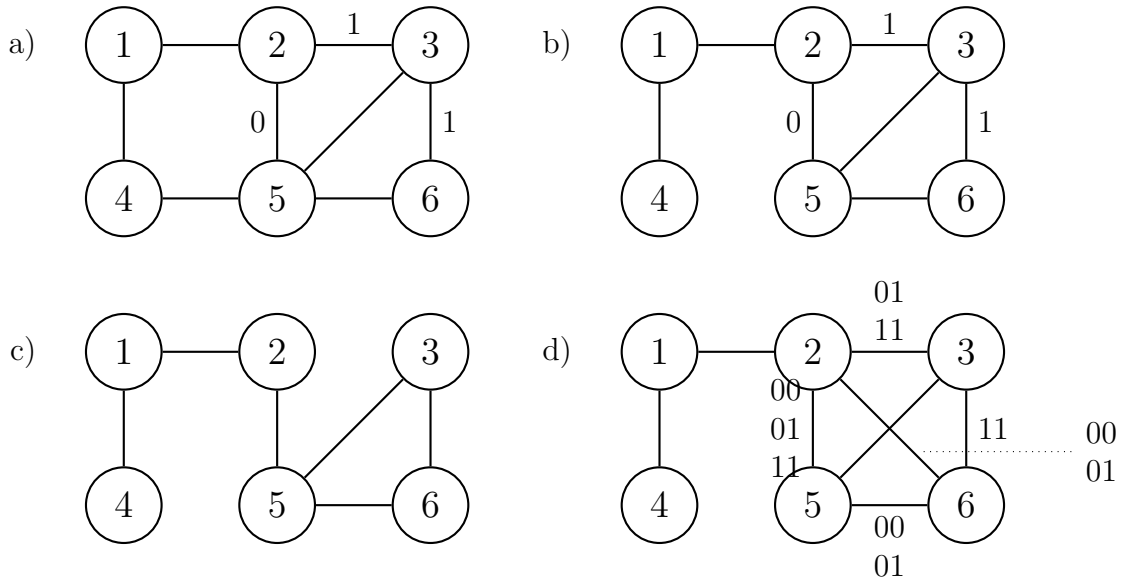

Figure 4.1: a) The generating model i.e the "correct" LGM. b) Graph usually discovered with $n \in [500, 1500]$ and $\kappa = 0.3$. c) Example of a graph discovered with $n = 500$ and $\kappa = 0.01$. d) Example of a graph discovered with $n = 500$ and $\kappa = 0.75$.

ally discover the correct structure with the correct labeling from $n = 500$ onwards save for the absence of the edge $4, 5$ which sometimes persists even to $n = 2000$. This anomaly can perhaps be explained by the fact that in our synthetic distribution, the parameter describing the effect between variables 4 and 5 is by accident relatively low. Thus the dependence between the variables is not strongly present in the data and our procedure which was designed to be a sceptical one to avoid overfitting doesn't recognize it. Note that 4 and 5 have no common neighbours and hence there can be no label between them and so adding a label on the edge for an outcome which may be rare in the data by chance brings no help. Conversely, the labeled edges with the correct labeling are more easily discovered since it is precisely the labeled edges i.e the context-specific dependencies that are reflected in the data.

The graphs in fig 4.1 c) and d) illustrate the significance of the κ-parameter and its choice. With low values, labels need to be very strongly supported by data to be included and with 0.01 are altogether absent as in fig 4.1 c). Also, labeled edges from the generating model start disappearing. This is precisely the reason to introduce the LGM in the first place: if there indeed exists a context-specific independence in the generating model, there might in the worst case be as much evidence against the correlation as their is for it. Fig. 4.1 d) on the other hand shows a cautionary example of the consequences of adding labels too freely. Because of inevitable noise in the data, false labels start showing

Variable	Explanation
X_1	Smoking
X_2	Strenuous mental work
X_3	Strenuous physical work
X_4	Systolic blood pressure > 140
X_5	Ratio of beta and alpha lipoproteins > 3
X_6	Family anamnesis of coronary heart disease

Table 4.1: Explanations for variables in the heart disease data

up along with false edges created by heavily labeling non-existent dependencies. Notice however, that the edge $4, 5$ is still absent: indeed the addition of labels doesn't remove the requirement for strong evidence for an edge in a situation where the edge cannot be labeled.

Next we look at a real data set consisting of 1841 observations of probable risk factors coronary thrombosis, the forming of a blood clot inside a blood vessel of the heart. Originally presented in Reinis et al. (1981) this data has since become somewhat of a classic in the field of graphical models (see for example Edwards and Havranek 1985, Whittaker 1990, Nyman et al. 2014). There are six binary variables, each corresponding to a yes-or-no -answer to a question about the presence of a coronary disease risk factor, for example if the individual has high blood pressure or not. All the variables are explained in Tab. 4.1. Fig. 4.2 shows the LGM with the highest posterior score for $\kappa = 0.3$ and $N = 1$. From the graph we can read that according to our model and the data, if we know for example that a person smokes, what kind of work he/she does tells us nothing about his/her blood pressure or cholesterol levels. Also, we could say that if he/she has an alarming lipoprotein ratio, smoking and blood pressure are independent factors. However, in the opposite case, these two factors are related. This data was previously studied by Nyman et al. (2014), who used decomposable LGMs (there called *stratified graphical models*). Their graph corresponds with ours save for ours having one edge less and one label more. This seems to indicate that our procedure adds dependencies more cautiously. We also note that our structure would not be possible under the decomposability requirement, a fact which could very well explain the differing label sets.

5 Discussion

Graphical models are useful, because they offer a way to represent complicated structures clearly and compactly. In the case of probability distributions, a graphical representation of the dependence structure immediately gives a general understanding of the relation-

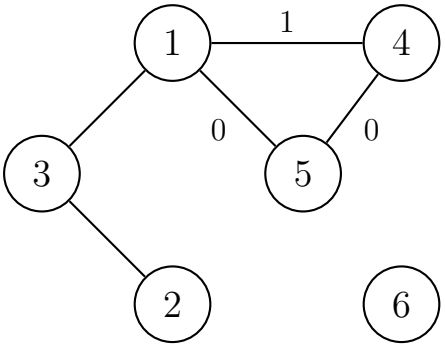

Figure 4.2: LGM model based on the heart disease data.

ships between the variables in a very intuitive way. It also facilitates the theoretical and computational manipulation of these structures by allowing to use graph theory and the associated methods.

The labelled graphical model is a way to increase the granularity of the graph model without losing the benefits of simplicity. On one hand it is more accurate in the sense that the model space is expanded to include the "in between" situations of dependence in some contexts and independence in others and on the other it can help discover the skeleton of the graph when direct dependencies are important to be included even if they are only present in a subset of outcomes.

The unavailability of the marginal likelihood for Markov networks and LGMs complicates the task of inferring these models using Bayesian methods. However, the pseudo-likelihood offers a way to approximate the likelihood in a way that still has a very clear interpretation in terms of the structure of the graph. Moreover, it is consistent as we have proven and doesn't require any tuning parameters. It also factorizes according to the structure of the graph, a feature that is very useful in practice for computational reasons.

In this work we have outlined the general theory for labelled graphical models and studied how it relates to two different parameterizations. We have presented the basics of Bayesian inference as it relates to graphical models and when faced with the problem of unavailable marginal likelihood, proposed a consistent approximation method: the marginal pseudolikehood, coupled with a proposal for prior distributions for both the nuisance parameters and the graph structure itself. We have then presented an algorithm that given a set of data, uses the pseudolikelihood based approximation as a score function to search for the LGM that best fits the data. Using both synthetic and real data we have studied the use of this algorithm in practice, pointing out possible pitfalls and problems as well as comparing our results with those presented in earlier research using different methods (see Nyman et al. 2014).

To keep the scope of this work sufficiently limited, the choice of the prior distribution

for model parameters was not thoroughly discussed. This could be an interesting topic for future research. Also, as mentioned, the algorithm used will become insufficient when the number of nodes grows and a more computationally efficient one is needed in the future. This is far from a trivial task and will be one of the goals of further development.

References

[Besag, 1972] Besag, J. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological).*

[Birch, 1963] Birch, M. (1963). Maximum likelihood in three-way contigency tables. *J. Roy. Statist. Soc.*

[Birch, 1964] Birch, M. (1964). Detection of partial association i: the 2x2 case. *J. Roy. Statist. Soc.*

[Bock, 1986] Bock, H.-H. (1986). Loglinear models and entropy clustering methods for qualitative data. *Classification as a tool of research (eds W. Gaul & M. Schader), 19-26.*

[Bock, 1994] Bock, H.-H. (1994). Information and entropy in cluster analysis. *Proceedings of the First US/Japan Conference on the Frontiers of Statistical Modeling: an informational approach (ed. H. Bozdogan), 115 147.*

[Bock, 1996] Bock, H.-H. (1996). Probability models and hypothesis testing in partitioning cluster analysis. *Clustering and classification (eds P. Arabie, L. Hubert & G. De Soete), 377453.*

[Buntine, 1991] Buntine, W. (1991). Theory refinement of bayesian networks. *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence.*

[Corander, 2003] Corander, J. (2003). Labelled graphical models. *Scandinavian Journal of Statistics.*

[Csiszár and Talata, 2006] Csiszár, I. and Talata, Z. (2006). Marginal pseudo-likelihood inference for markov networks. *Annals of Statistics.*

[Darroch et al., 1980] Darroch, J., Lauritzen, S., and Speed, T. (1980). Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics.*

[Edwards and Havranek, 1985] Edwards, D. and Havranek, T. (1985). A fast procedure for model search in multi-dimensional contingency tables. *Biometrika.*

[Gelman et al., 2013] Gelman, A., Carlin, J., Hal, S., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis.* Chapman and Hall.

[Hø jsgaard, 1998] Hø jsgaard, S. (1998). *Split models for contingency tables.* PhD thesis, Danish Institute of Agricultural Science, Biometry Research Unit, Tjele, Denmark.

[Hø jsgaard, 2003] Hø jsgaard, S. (2003). Statistical inference in context specific interaction models for contingency tables. *Scand J Statist.*

[Kass and Wasserman, 1996] Kass, R. E. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association.*

[Koller and Friedman, 2009] Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques.* MIT Press.

[Lauritzen, 1996] Lauritzen, S. L. (1996). *Graphical Models.* Oxford University Press.

[MacKay, 2003] MacKay, D. (2003). *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press.

[Nyman et al., 2014] Nyman, H., Pensar, J., Koski, T., and Corander, J. (2014). Stratified graphical models - context-specific independence in graphical models. *Bayesian Analysis.*

[Pensar et al., 014a] Pensar, J., Nyman, H., Koski, T., and Corander, J. (2014a). Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models. *Data Mining and Knowledge Discovery.*

[Pensar et al., 014b] Pensar, J., Nyman, H., Niiranen, J., and Corander, J. (2014b). Marginal pseudo-likelihood inference for markov networks. *arXiv:1401.4988.*

[Ravikumar et al., 2010] Ravikumar, P., Wainwright, M. J., and Lafferty, J. (2010). High-dimensional ising model selection using l_1-regularized logistic regression. *Annals of Statistics.*

[Reinis, 1981] Reinis, Z. e. a. (1981). Prognostic significance of the risk profile in prevention of coronary heart disease. *Bratis. Lek. Listy.*

[Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics.*

[Silander et al., 2007] Silander, T., Kontkanen, P., and Myllymki, P. (2007). On sensitivity of the map bayesian network structure to the equivalent sample size parameter. *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence.*

[Tsamardinos et al., 2003] Tsamardinos, I., Aliferis, C., Statnikov, A., and Statnikov, E. (2003). Algorithms for large scale markov blanket discovery. *The 16th International FLAIRS Conference, St.*

[Whittaker, 1990] Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. Wiley.

Appendix A.

Here we investigate the asymptotic behaviour of the logMPL. For one variable, we have

$$\log p(\mathbf{X}_j|G_L) = \\ \sum_{l=1}^{k_j}\left[\log\Gamma\left(\alpha_{jl}\right) - \log\Gamma\left(n_{jl}+\alpha_{jl}\right) + \sum_{i=1}^{r_j}\left(\log\Gamma\left(n_{ijl}+\alpha_{ijl}\right) - \log\Gamma\left(\alpha_{ijl}\right)\right)\right].$$

First we note that we can ignore

$$\sum_{l=1}^{k_j}\left[\log\Gamma\left(\alpha_{jl}\right) - \sum_{i=1}^{r_j}\log\Gamma\left(\alpha_{ijl}\right)\right]$$

since it does not depend on n. Then applying Stirling's formula

$$\log p(\mathbf{X}_j|G_L) = \\ \sum_{l=1}^{k_j}\Bigg[-(n_{jl}+\alpha_{jl}-\frac{1}{2})\log(n_{jl}+\alpha_{jl}) + (n_{jl}+\alpha_{jl}) \\ + \sum_{i=1}^{r_j}\left[(-(n_{ijl}+\alpha_{ijl}-\frac{1}{2})\log(n_{jl}+\alpha_{ijl}) + (n_{ijl}+\alpha_{ijl})\right]\Bigg] + O(1).$$

We note that $\sum_{i=1}^{r_j} n_{ijl} = n_{jl}$ and thus

$$\begin{aligned}\log p(\mathbf{X}_j|G_L) = &\sum_{l=1}^{k_j}\sum_{i=1}^{r_j} n_{ijl}\log\frac{n_{ijl}+\alpha_{ijl}}{n_{jl}+\alpha_{jl}} \\ &+ \sum_{l=1}^{k_j}\sum_{i=1}^{r_j}\alpha_{ijl}\log\frac{n_{ijl}+\alpha_{ijl}}{n_{jl}+\alpha_{jl}} \\ &+ \sum_{l=1}^{k_j}\left[\frac{1}{2}\log(n_{jl}+\alpha_{jl}) - \sum_{i=1}^{r_j}\frac{1}{2}\log(n_{ijl}+\alpha_{ijl})\right] \\ &+ O(1).\end{aligned}$$

Because

$$\frac{n_{ijl}+\alpha_{ijl}}{n_{jl}+\alpha_{jl}} \to \frac{n_{ijl}}{n_{jl}} \quad \text{as} \quad n\to\infty,$$

we have for the first term

$$\sum_{l=1}^{k_j}\sum_{i=1}^{r_j} n_{ijl}\log\frac{n_{ijl}+\alpha_{ijl}}{n_{jl}+\alpha_{jl}} = \sum_{l=1}^{k_j}\sum_{i=1}^{r_j} n_{ijl}\log\frac{n_{ijl}}{n_{jl}}.$$

The second term we omit as constant and for the final term we have

$$\begin{aligned}
&\sum_{l=1}^{k_j}\left[\frac{1}{2}\log(n_{jl}+\alpha_{jl}) - \sum_{i=1}^{r_j}\frac{1}{2}\log(n_{ijl}+\alpha_{ijl})\right] \\
&= \frac{1}{2}\sum_{l=1}^{k_j}\left[\log\frac{n_{jl}+\alpha_{jl}}{n} - \log n - \sum_{i=1}^{r_j}\left[\log\frac{n_{ijl}+\alpha_{ijl}}{n} - \log n\right]\right] \\
&= \frac{1}{2}\sum_{l=1}^{k_j}\left[-\log n + \sum_{i=1}^{r_j}\log n\right] + \frac{1}{2}\sum_{l=1}^{k_j}\left[\log\frac{n_{jl}+\alpha_{jl}}{n} - \sum_{i=1}^{r_j}\log\frac{n_{ijl}+\alpha_{ijl}}{n}\right] \\
&= \frac{1}{2}\sum_{l=1}^{k_j}\left[-\log n + \sum_{i=1}^{r_j}\log n\right] + O(1) \\
&= \frac{1}{2}(-k_j\log n + k_j r_j\log n) + O(1) \\
&= \frac{k_j(r_j-1)}{2}\log n + O(1).
\end{aligned}$$

Gathering all this we conclude that as $n \to \infty$ the logMPL estimator for one variable is asymptotically equal to

$$\sum_{l=1}^{k_j}\sum_{i=1}^{r_j} n_{ijl}\log\frac{n_{ijl}}{n_{jl}} - \frac{(r_j-1)k_j}{2}\log n.$$

Noteworthy is that if the structure is an unlabelled one, this is equal to the expression Pensar et al. (2014b) arrived at for the normal logMPL which further corresponds to the PIC estimator of Csiszár and Talata (2006). Moreover, when labels are added the only differing element is the count of outcome configurations k_j. The first term for ordinary Markov networks is the logarithm of the maximum pseudolikelihood as defined by Csiszár and Talata (2006) and can be regarded as the one approximating the likelihood. The second term is a penalty term which regulates overfitting.

Appendix B.

The algorithm for learning a LGM from data set $\mathbf{X}$ as pseudocode. Note that since the underlying graph $G = (V, E)$ will usually be represented by a single data structure, for

example an adjacency-matrix, no variable explicitly refers to the set of nodes V. Instead E is understood here as a structure that automatically comprises the idea of a graph with a specific number of nodes.

findLGM(data set $\mathbf{X}$, set of possible edges E_{pos})
1 E = empty
2 L = empty
3 *continue* = TRUE
4 **while** *continue*
5 *continue* = FALSE
6 $imp = 0$
7 $E_{cand} = E$
8 $L_{cand} = L$
9 **for all** $\{i, j\}$ in E_{pos}
10 $E_{temp} = E$
11 **if** $\{i, j\}$ is in E
12 add $\{i, j\}$ to E_{temp}
13 **else**
14 remove $\{i, j\}$ from E_{temp}
15 L_{temp} = **optimizeLabels**($\mathbf{X}$, E_{temp}, L_{temp})
16 imp_{temp} = **logMPL**($\mathbf{X}$, E, L) - **logMPL**($\mathbf{X}$, E_{temp}, L_{temp})
17 **if** $imp_{temp} > imp$
18 $imp = imp_{temp}$
19 $E_{cand} = E_{temp}$
20 $L_{cand} = L_{temp}$
21 **if** $imp > 0$
22 $E = E_{cand}$
23 $L = L_{cand}$
24 *continue* = TRUE
25 return (E, L)

optimizeLabels(data set $\mathbf{X}$, set of edges E, labelset L**)**

1 for all $\{i, j\}$ in E
2 $continue$ = TRUE
3 **while** $continue$
4 $continue$ = FALSE
5 $L_{cand} = L$
6 $imp = 0$
7 **for all** $x_{P_{\{i,j\}}}$ in $\mathcal{X}_{P_{\{i,j\}}}$
8 **if** $x_{P_{\{i,j\}}}$ is not in $L_{\{i,j\}}$ AND $\{x_{P_{\{i,j\}}} \cup L_{\{i,j\}}\} \subset \mathcal{X}_{P_{\{i,j\}}}$
9 $L_{temp} = L$
10 add $x_{P_{\{i,j\}}}$ to $L_{temp_{\{i,j\}}}$
11 make L_{temp} maximal and regular
12 imp_{temp} = **logMPL(X**, E, L**)** - **logMPL(X**, E, L_{temp}**)**
13 **if** $imp_{temp} > imp$
14 $imp = imp_{temp}$
15 $L_{cand} = L_{temp}$
16 **else**
17 $L_{temp} = L$
18 remove $x_{P_{\{i,j\}}}$ from $L_{temp_{\{i,j\}}}$
19 imp_{temp} = **logMPL(X**, E, L**)** - **logMPL(X**, E, L_{temp}**)**
20 make L_{temp} maximal and regular
21 **if** $imp_{temp} > imp$
22 $imp = imp_{temp}$
23 $L_{cand} = L_{temp}$
24 **if** $imp > 0$
25 $L = L_{cand}$
26 $continue$ = TRUE
27 return L